Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method comprising:

providing a test hierarchy implemented in a common source unit including production source code and test source code, the test hierarchy including a collection of test methods, ~~each~~a test method in the collection of test methods including test source code specifying a call[[ing]] to a test assertion method[[s]] for ~~checking correctness of~~ testing the production source code, ~~each~~ the test assertion call defining a test execution flow within the test hierarchy in an event of a failure; and

controlling the test execution flow inside the test hierarchy in the event of failure according to a parameter value of the call to the test assertion method,~~call in the event of the failure~~ the parameter values available for controlling the test execution flow including values specifying:

continuing the test execution flow with a next statement in the test method;

interrupting the test execution flow of the test method and continuing the test execution flow with any remaining test methods in the test heirarchy; and

interrupting the test execution flow of a test class containing the test method and continuing the test execution flow with any remaining test classes in the test heirarchy.

2. (Original) The method of claim 1 wherein the parameter causes instructions to continue or to abort at any level of the test hierarchy after the event.

3. (Currently Amended) The method of claim 1 wherein a test assertion method includes instructions for verifying an expected state of the production source code.

4. (Currently amended) The method of claim 1 wherein the test hierarchy is implemented using an object oriented programming language including at least one of the ABAP ~~(advanced business application program language)~~ programming language, ~~Java~~ JAVA programming language, C++ programming language, ~~and~~ or C# programming language.

5. (Original) The method of claim 1 wherein the test hierarchy is implemented using a procedural programming language including at least one of C programming language, Fortran programming language and Pascal programming language.

6. (Original) The method of claim 1 wherein the test hierarchy is implemented in a unit test environment.

7. (Currently amended) A computer program product, tangibly embodied in ~~an information carrier~~ a machine-readable storage device, the computer program product comprising instructions operable to:

provide a test hierarchy implemented in a common source unit including production source code and test source code, the test hierarchy including a collection of test methods, ~~each~~ a test method in the collection of test methods including test source code specifying a call[[ing]] to a test assertion method[[s]] for ~~checking the correctness of~~ testing the production source code, ~~each~~ the test assertion call defining a test execution flow inside the test hierarchy in an event of failure detection; and

control the test execution flow within the test hierarchy in the event of failure according to a parameter value of the call to the test assertion method, ~~call in response to detecting the event~~ the parameter values available for controlling the test execution flow including values specifying:

continuing the test execution flow with a next statement in the test method;

interrupting the test execution flow of the test method and continuing the test execution flow with any remaining test methods in the test heirarchy; and

interrupting the test execution flow of a test class containing the test method and continuing the test execution flow with any remaining test classes in the test heirarchy.

8. (Original) The product of claim 7 wherein the parameter causes the test execution flow to continue or to abort at any level of the test hierarchy after the occurrence of the event.

9. (Currently Amended) The product of claim 7 wherein a test assertion method includes instructions for verifying an expected state of the production source code.

10. (Currently amended) The product of claim 7 wherein the test hierarchy is implemented using an object oriented programming language including at least one of the ABAP ~~(advanced business application program language)~~ programming language, ~~Java~~ JAVA programming language, C++ programming language, ~~and~~ or C# programming language.

11. (Original) The product of claim 7 wherein the test hierarchy is implemented using a procedural programming language including at least one of C programming language, Fortran programming language and Pascal programming language.

12. (Original) The product of claim 7 wherein the test hierarchy is implemented in a unit test environment.

13. (Currently amended) A computer system comprising:

a processor for executing program instructions; and

one or more memory devices storing program instructions implementing a test hierarchy implemented in a common source unit including production source code and test code, the test hierarchy including a collection of test methods, ~~each~~ a test method in the collection of test methods including test source code specifying a call[[ing]] to a test assertion method[[s]] for ~~checking the correctness of~~ testing the production source code, ~~each~~ the test assertion call defining the test execution flow within the test hierarchy in an event of a failure, [[; and]]

~~a means for controlling~~the one or more memory devices further storing program instructions operable to control a flow inside the test hierarchy in the event of failure according to a parameter value of the call to the test assertion method, ~~call in response to the event~~ the parameter values available for controlling the test execution flow including values specifying:

continuing the test execution flow with a next statement in the test method;

interrupting the test execution flow of the test method and continuing the test execution flow with any remaining test methods in the test heirarchy; and

interrupting the test execution flow of a test class containing the test method and continuing the test execution flow with any remaining test classes in the test heirarchy.

14. (Currently Amended) The system of claim 13 wherein ~~the test assertion is called with a~~ the parameter ~~causing~~ causes the test execution to continue or to abort at any level of the test hierarchy after the event.

15. (Currently Amended) The system of claim 13 wherein a test assertion method includes instructions for verifying an expected state of the production source code.

16. (Currently amended) The system of claim 13 wherein the test hierarchy is implemented using an object oriented programming language including at least one of the ABAP ~~(advanced business application program language)~~ programming language, ~~Java~~ JAVA programming language, C++ programming language, ~~and~~ or C# programming language.

17. (Original) The system of claim 13 wherein the test hierarchy is implemented using a procedural programming language including at least one of C programming language, Fortran programming language and Pascal programming language.

18. (Original) The system of claim 13 wherein the test hierarchy is implemented in a unit test environment.

19. (New) The method of claim 1, wherein the parameter values available for controlling the test execution flow include a value specifying:

interrupting the test execution flow of a program frame containing the test method and continuing the test execution flow with any remaining program frames in the test hierarchy.

20. (New) The product of claim 7, wherein the parameter values available for controlling the test execution flow include a value specifying:

interrupting the test execution flow of a program frame containing the test method and continuing the test execution flow with any remaining program frames in the test hierarchy.

21. (New) The system of claim 13, wherein the parameter values available for controlling the test execution flow include a value specifying:

interrupting the test execution flow of a program frame containing the test method and continuing the test execution flow with any remaining program frames in the test hierarchy.

Amendments to the Drawings:

The attached replacement sheets of drawings FIGs. 1-6 replace the original sheets including FIGs. 1-6.